VIVO

enabling national
networking of scientists

Managing Your Data Flows: Architecture and Data Provenance For Your Institution

John Fereira
Cornell University

Google Drive

- https://goo.gl/Nz1Ycs

Managing your data Workflows = managing data ingest: There are many apps for that

- John: VIVO Harvester
- Alex: Integrating Symplectic Elements with the harvester
- Violeta: Using Karma for data ingest

Some other tools

- Google Refine
- Custom tools
    - R Tools (Mike Conlon)
    - Python tools (Ted Lawless)
    - Generalized XSLT based RDF Ingest (Joe McEnerney)

Data Ingest: The Easy way

Use the VIVO UI to add and edit data

- Good for smaller institutions
- Employ staff or students for data entry
- Good for prototyping system
- https://wiki.duraspace.org/display/VIVO/Manual+Data+Entry

Types of Data Sources

- Human Resources data
- Faculty reporting systems
- Course information from registrars office
- Publications (Web of Science, Pubmed…)
- Grants data
- Event calendars
- Data can be simple, but it must be structured

VIVO Harvester: what is it

- A collection of tools, encapsulated as a jar file
- A set of examples for each of the tools
- A set of “full harvest” examples
- It is not a turn-key solution
- Complete harvest uses a pipeline of tools
- Custom tools can be inserted into the pipeline
- Tools can be integrated with apps such as Symplectic harvester, Karma, Google Refine…

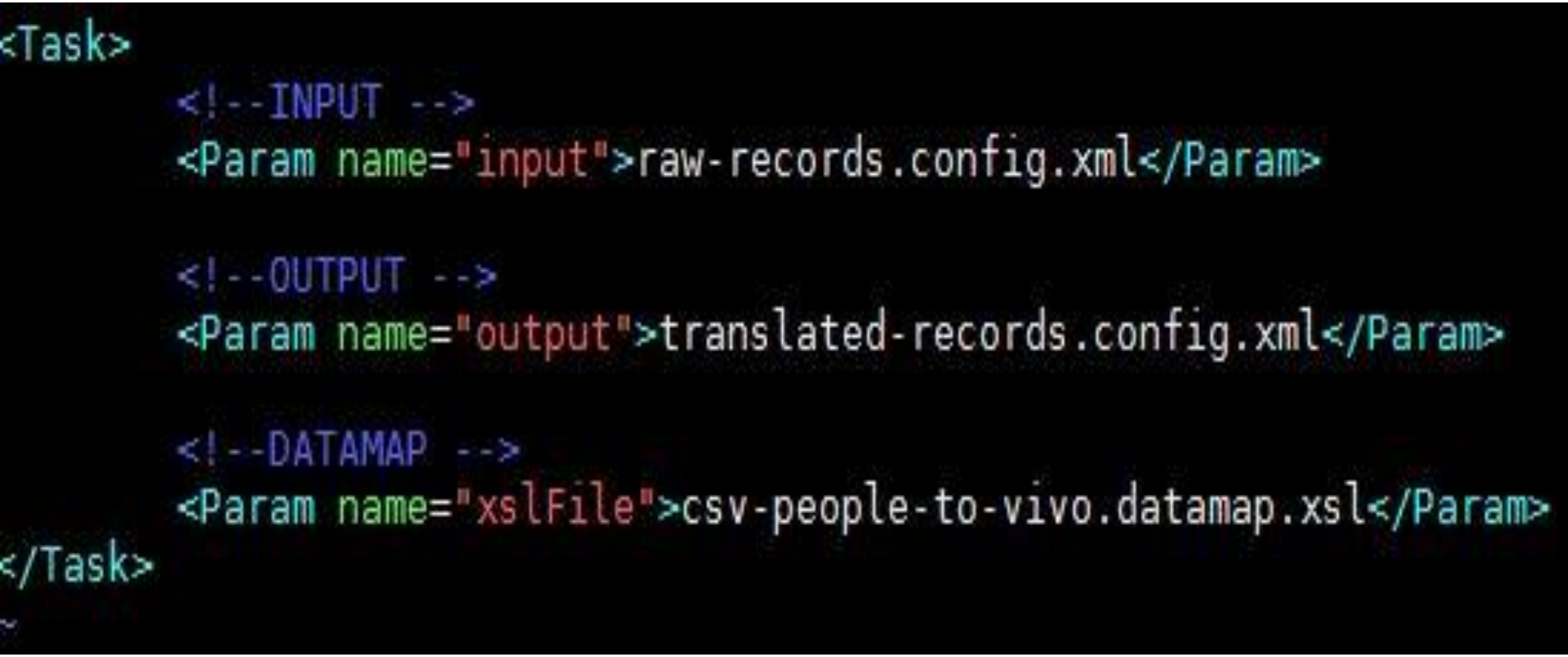
VIVO Harvester: how to get it

- Go to some directory (e.g. /usr/local)
- git clone git@github.com:vivo-project/VIVO-Harvester.git
- cd VIVO-Harvester
- Checkout develop branch
  - git checkout develop
- Build jar file (requires maven)
  - mvn clean dependency:copy-dependencies package -DskipTests=true

Tools have a common invocation pattern

- Call a java class with arguments
- Can specify each argument individually or in a CONFIG file
- All tools have a wrapper script
  - bin/harvester-jdbcfetch
  - bin/harvester/transfer
- Set classpath, set JAVA Options, invoke class
- wrapper scripts set min/max memory to 1gb

Configuration file example



```
<Task>
        <!--INPUT -->
        <Param name="input">raw-records.config.xml</Param>

        <!--OUTPUT -->
        <Param name="output">translated-records.config.xml</Param>

        <!--DATAMAP -->
        <Param name="xslFile">csv-people-to-vivo.datamap.xsl</Param>
</Task>
~
```

Record Handlers

- Most tools have input and output artifacts
- Format of artifacts specified using a record handler
- Record handlers specified using xml config file
- Input and Output record handlers do not need to be the same
- Start development with TextFileRecordHandler, use other RecordHandlers for better performance

Type of Record Handlers

- TextFileRecordHandler
  - One file for each “record”
- JenaRecordHandler
  - Supports sdb, tdb, rdb
- JDBCRecordHandler
  - Uses H2 database by default
- MapRecordHandler
  - Stores records in memory as a map

A few useful utilities

- CSVtoJDBC – loads a CSV file into a database (H2 is the default)
- DatabaseClone – copy from external database into local instance
- JenaConnect – execute sparql against a triple store, clear triples in a model, good for debugging.

Harvesting Pipeline

- Fetcher/Parser
- Translate: maps parsed data to “vivo” RDF
- Transfer to local triple store (Jena TDB)
- Disambiguate using Scoring/Matching
- Changenamespace (mint unique URIs)
- Diff with previous model to create subtractions and additions
- Transfer rdf to VIVO triple store

Fetching and Parsing

- Fetches data from a URL, Database, local file
- Many different types of fetchers
  - JDBCFetch
  - SimpleXMLFetch
  - JSONFetcher
  - OAIFetch
  - D2RMapFetch
  - PubmedFetch
  - CSVFetch (superceded by CSVToJDBC, CSVtoRDF)
  - LinkedDataHarvester (load data from another VIVO instance)

Output of Fetcher

- Creates “fake” namespace
- Namespace prefix uses for mapping fields
- Namespace matched for disambiguation and minting “real” URIs

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:node-person="http://vivo.example.com/harvest/aims_users/fields/person/"
         xml:base="http://vivo.example.com/harvest/aims_users/person">
  <rdf:Description rdf:ID="node_-_0">
    <rdf:type rdf:resource="http://vivo.example.com/harvest/aims_users/types#person"/>
    <node-person:Picture>http://aims.fao.org/sites/default/files/profiles/profile_image_108074.jpg</node-person:Picture>
    <node-person:Website>http://www.valeriapesce.name</node-person:Website>
    <node-person:Nid>108074</node-person:Nid>
    <node-person:Profile>In the last six years at the Global Forum on Agricultural research (GFAR) I have worked extensively on
metad
ata standards and protocols for managing and exchanging information between systems, in strict collaboration with the OEKCS
group in
FAO.</node-person:Profile>
    <node-person:Organization>Food and Agriculture Organization of the United Nations (FAO)</node-person:Organization>
    <node-person:Expertise>Information management tools, information systems, information architectures</node-person:Expertise>
    <node-person:LastName>Pesce</node-person:LastName>
    <node-person:Country>Italy</node-person:Country>
    <node-person:Email>valeria.pesce@fao.org</node-person:Email>
    <node-person:geolocation>http://aims.fao.org/aos/geopolitical.owl#Italy</node-person:geolocation>
    <node-person:Profile_URL>http://aims.fao.org/node/108074</node-person:Profile_URL>
    <node-person:Username>valeria.pesce</node-person:Username>
    <node-person:FirstName>Valeria</node-person:FirstName>
    <node-person:Role>Information Management Specialist</node-person:Role>
    <node-person:Interests>agINFRA, AgriDrupal, AgriFeeds, AgriVIVO, authority control, automatic indexing, CIARD Content
Management
Task Force, CIARD RING, cloud services, CMS - Content Management Systems, data exchange, Drupal, IAALD - International
Association of
 Agricultural Information Specialists, information management, institutional repository software, interoperability, Linked Open Data
- LOD, RDF - Resource Description Framework, Semantic Web</node-person:Interests>
  </rdf:Description>
</rdf:RDF>
```

XSLTranslator

- Map “fake” namespace to VIVO classes and properties
- Uses XSLT transform
- Unique ID for each record
- node-person:Organization becomes foaf:Organization
- Relationships created

Translated RDF

```
<rdf:Description rdf:about="http://vivo.example.com/harvest/aims_users/person/uid-108074">
    <rdf:type rdf:resource="http://xmlns.com/foaf/0.1/Person"/>
    <rdfs:label>Pesce, Valeria</rdfs:label>
    <core:currentMemberOf rdf:resource="http://vivo.example.com/harvest/aims_users/org/aims"/>
    <foaf:firstName>Valeria</foaf:firstName>
    <foaf:lastName>Pesce</foaf:lastName>
    <core:primaryEmail>valeria.pesce@fao.org</core:primaryEmail>
    <core:positionInOrganization rdf:resource="http://vivo.example.com/harvest/aims_users/org/Food%20and%20Agriculture%20Organization%20of%20the%20United%20Nations%20(FAO)"/>
  </rdf:Description>

  <rdf:Description rdf:about="http://vivo.example.com/harvest/aims_users/org/Food%20and%20Agriculture%20Organization%20of%20the%20United%20Nations%20(FAO)">
    <rdf:type rdf:resource="http://xmlns.com/foaf/0.1/Organization"/>
    <rdfs:label>Food and Agriculture Organization of the United Nations (FAO)</rdfs:label>
    <core:organizationForPosition rdf:resource="http://vivo.example.com/harvest/aims_users/position/positionFor108074inFood%20and%20Agriculture%20Organization%20of%20the%20United%20Nations%20(FAO)"/>
    <core:hasGeographicLocation rdf:resource="http://aims.fao.org/aos/geopolitical.owl#Italy"/>
  </rdf:Description>
```

Transfer

- Load RDF into TDB triplestore
- Duplicate URIs are not loaded
- Further operations are made in the triple store

Scoring/Match

- Disambiguates People, Organizations, etc. based upon property values
- Supports Equality, NameCompare, NormalizedLevenshteinDifference, Soundex algorithms
- Each property is weighted
  - firstName: 0.5
  - lastName: 0.5
  - Email: 1.0
- MatchThreshHold: 1.0

Matching

- Determines what should be done with a record which matches another record based upon it’s “score”
  - Replace old record
  - Merge records
  - Ignore record

ChangeNameSpace

- Match old namespace pattern in configuration file

http://vivo.example.com/harvest/aims_users/person/

- Specify namespace in VIVO

http://agrivivodev.mannlib.cornell.edu/vivo/individual/

- Mint a new URI in the vivo namespace

http://agrivivodev.mannlib.cornell.edu/vivo/individual/n123456

- Optionally created sameAs statement (useful for when ingesting from another VIVO instance)

Diff of previous harvest

- Compare TDB model with previous harvest
- Generate vivo-additions.rdf
- Generate vivo-substractions.rdf
- DEMO

Final Transfer

- Load vivo-subtractions.rdf file into VIVO triple store
- Load vivo-additions.rdf file into VIVO triple store

Integration with VIVO web services

- ListRDF
- SparqlQuery
  - Requires vivo 1.7+
- SparqlUpdate
  - Requires vivo 1.6
  - Can replace final transfer
  - No Need for reindexing or inferencing!

Putting it all together

- example-scripts/bash-scripts/full-harvest-examples
  - 1.0-1.5-examples
  - 1.6-examples (needs development)
- Use demodb, vitrodb_test in VIVO-Harvester/support directory

Thank you